

개발 완료 보고서

제출일 : 2024년 3월 27일

팀명	8 팀		
참여 인원	김승수, 권철민, 유시온		
개발프로젝트 소개			
프로젝트 명	‘복겟 몬스타’		
활동 일시	‘24. 03. 22. ~ ‘24. 03. 27	장소	광주인력개발원 공학1관 2층 드론융합실
주요 주제	C 언어의 반복문, 제어문, 랜덤함수(난수생성)을 이용한 프로그램 개발		
개발 목적	C 언어의 함수, 반복문, 선택문, 제어문, 랜덤함수(난수생성)을 이용한 프로그램 개발		
파 트 별 임무 분담	김승수	순서도 작성 및 전투로직 구현	
	권철민	요구분석서 작성 및 메인, 아이템 로직 구현	
	유시온	문서 작성 및 맵 이동 로직 구현, 총괄	
개발 환경	실습실 / 개인pc 사용 / Ubuntu 16.04 Linux os / Visual Studio Code 활용		

사용함수	void show_line(char ch, int num) : 상태창 라인 줄 표시	
	void enter(void) : 버퍼에 개행 문자 외 남는 문자 제거	
	void cal_exponent(int a, int* b) : a의 자리수를 b에 저장	
	int getch(void) : window에 사용 가능 한 getch 쓰기 위한 함수	
변수(명주석)	int hp : 유저 현재 체력	int item[10][1] : 아이템 배열
	int hp_max : 유저 최대 체력	int input : 아이템 사용시 입력 값
	int level : 레벨	char name[27][10] : 반 이름 배열
	int monster_hp : 몬스터 현재 체력	char username[10] : 유저 이름 입력시 배열
	int monster_randnum : 몬스터 등장 여부	char monster_name[10] : 몬스터 이름 배열
	char user : 맵 상 유저 이동 표시	int monster_hp_max : 몬스터 최대 체력
	char map[YY][XX] : 맵 이중배열	int attack_user : 유저 공격력
	float randnum : 랜덤 숫자	int attack_mon : 몬스터 공격력
	int half_use : 절대무적 사용 여부	int flag : 도망 여부
	int same : 원소와 원소 비교시 사용	int choice : 전투 상태창 선택 값
	char key : 맵 이동 입력 값	int result_battle : 전투 결과
	int temp_hp : 이전 체력 임시 보관	char bok[10] : 복이 이름 배열

항 목		03/22	03/23	03 / 24	03 / 25	03 / 26	03 / 27
회의	개발계획서 및 일정표 작성						
	중간점검						
구현	메인						
	로직 / 전투						
	로직 / 맵 구현						
	로직 / 엔딩						
	테스트 및 디버깅						
	완료보고서 및 문서 작성						
제출							

요구분석서

요구사항 분석서

유형	분류	설명	
		발생조건/ 세부분류	세부내용
시작	시작	유저이름 입력	scanf() 이름 입력
맵	맵	맵 출력	7*7 이차원 배열로 맵 출력
		맵 구현	a,s,d,w 입력 시 이동 표현
		몬스터 만남	랜덤 숫자(0,1) 로 1이면 몬스터 만남 표현
			복이 인지 아닌지 판단
전투	몬스터 전 투	전투화면	1턴 당 공격, 아이템, 도망 선택 가능
		도 망	확률 30% , 실패 시 몬스터에게 공격 당하고 1턴 끝남
		유저 공격력	(유저 현재 체력/2) * (1~4배)
		몬스터 이름	유저 이름과 중복되지 않은 이름 출력
		몬스터 체력	유저 최대 체력 * (1~3배)
		몬스터 공격력	유저 최대 체력 * (0.5~3배)
		전투 승리	체력 회복 포션 0~10개 랜덤 지급(최대 보유 개수 300개)
			유저 최대 체력 *(120%~250%) 상승 후 재 산정
			레벨 증가
	복 이 전 투	복이 체력	유저 최대 체력 * (5~8배)
		전투 반복	복이 체력 1 이상 일시
종료	엔 딩	배드 엔딩	유저 체력 0 이하
		해피 엔딩	복이 체력 0 이하

코드설명	[랜덤 숫자 seed 초기화] srand(time(NULL));	
	[맵 이중 배열 생성] for(y=0; y<YY; y++){ for(x=0;x<XX;x++){}}	: 이중 for문 사용하여 생성
	[유저이름 입력] while(1){ printf("이름을 입력해주세요: "); scanf("%s", username); enter(); for(int i = 0; i < 27; i++){ same = 0; for(int j = 0; j < 10; j++){ if(name[i][j]==username[j]){ ++same; } } if(same == 10){ for(int k = 0; k < 10; k++){ name[i][k] = bok[k]; break; } } if(same == 10) break; printf("이름을 다시 입력하여 주시길 바랍니다.\n"); } }	:유저 이름 입력 :유저 이름이 반이름 배열에 있다면 복리로 교체 (same이 10일 때) : 교체 완료시 break 문으로 for문 종료 : 교체 완료시 break 문으로 while문 종료 : 배열에 없는 이름이면 while문으로 돌아가서 다시 이름 입력 받기
	[게임 반복] While(1) { [맵 출력] for(y=0; y<YY; y++){ for(x=0;x<XX;x++){ printf("%c",map[y][x]); printf("\n");}	: while 문 조건을 1로 해서 엔딩 시 break문으로 나오도록 함 : 이중 for문으로 맵 출력

코드설명

[이동키 입력]
printf("wasd로 이동 아이템창 i₩n");
key = getch();

while(key != 'a' && key != 's' &&
key != 'd' && key != 'w'&&
key != 'i'){
key = getch(); }

: getch()로 이동키 입력 받음

: while문 이용해서 a,s,w,d(이동키),
i(아이템 사용키) 아니면 다시
입력하도록 설정

[맵 이동]
for(y=0; y<YY; y++){
for(x=0;x<XX;x++){
if(map[y][x]==user){
if(key == 'a' && x > 4){
map[y][x]=' ';
map[y][x-3]=user;
break; }
else if(key == 's' && y < YY-2){
map[y][x]=' ';
map[+y][x]=user;
break; }
else if(key == 'd' && x < XX-5){
map[y][x]=' ';
map[y][x+3]=user;
break; }
else if(key == 'w' && y > 1){
map[y][x]=' ';
map[y-1][x]=user;
break; } } } }

: 'a' 누르면 x가 4보다 컸을 때
x 3만 큼 감소

: 's' 누르면 y가 YY(20)-2 보다
작았을 때 y 1 증가

: 'd' 누르면 x가 XX(60)-5 보다
작았을 때 x 3 증가

: 'w' 누르면 y가 1보다 컸을 때
x 1만 큼 감소

[몬스터 등장 여부]
monster_randnum = rand()%2;

: 랜덤 숫자가 1이면 몬스터 만남

[몬스터 랜덤 이름]
randnum = rand()%27;
char monster_name[10];
if(level == 5) {
for(int i = 0; i < 10; i++)
monster_name[i] = bok[i]; }
else {
for(int i = 0; i < 10; i++)
monster_name[i] =
name[(int)randnum][i]; }

: 랜덤 숫자 생성해서 배열에 적용

: (레벨이 5이면(전투에서 5번 승리시)
복이를 만나게 되는 조건)->
몬스터 복이로 생성

: 몬스터 랜덤 생성(랜덤숫자 적용)

코드설명

```
[몬스터 체력 생성]
if(same == 10)
    randnum = rand()%4 + 5;
else
    randnum = rand()%3 + 1;

monster_hp = hp_max * randnum;
int monster_hp_max = monster_hp;
```

: (same == 10) 배열에 이름이
복이일 경우 조건
복이체력 유저 최대체력 * (5~8배)

: 아닐시 일반 몬스터 체력 조건
몬스터 체력 유저 최대체력 * (1~3배)

```
[전투 상태창 선택- 1)싸우기]
switch(choice) {
case 1: {
randnum =(rand() % 4) + 1;
attack_user =
    (int)hp * 0.5 * randnum;
monster_hp -= attack_user;
flag = 1;
break; }
```

: 랜덤 숫자 생성

: 유저 공격력 생성
남은 체력 * (1~4배)

```
[전투 상태창 선택- 2)아이템, 체력포션]
case 2: {
scanf("%d", &input);
switch(input) {
case 1: {
randnum =(rand() % 51) + 30;
temp_hp = hp;
hp +=(int)((float)hp_max
*((float)randnum/100));
item[0][0]--; }
```

: 랜덤숫자 생성해서 체력 회복

```
[전투 상태창 선택- 2)아이템, 절대무적]
case 2: {
half_use = 1;
item[1][0]--;
break; }
```

: half_use를 1로 변경해서 전투시
체크해서 데미지 절반으로 줄여줌

: 몬스터에게 공격 당했을 시 조건문
확인 후 체력 차감

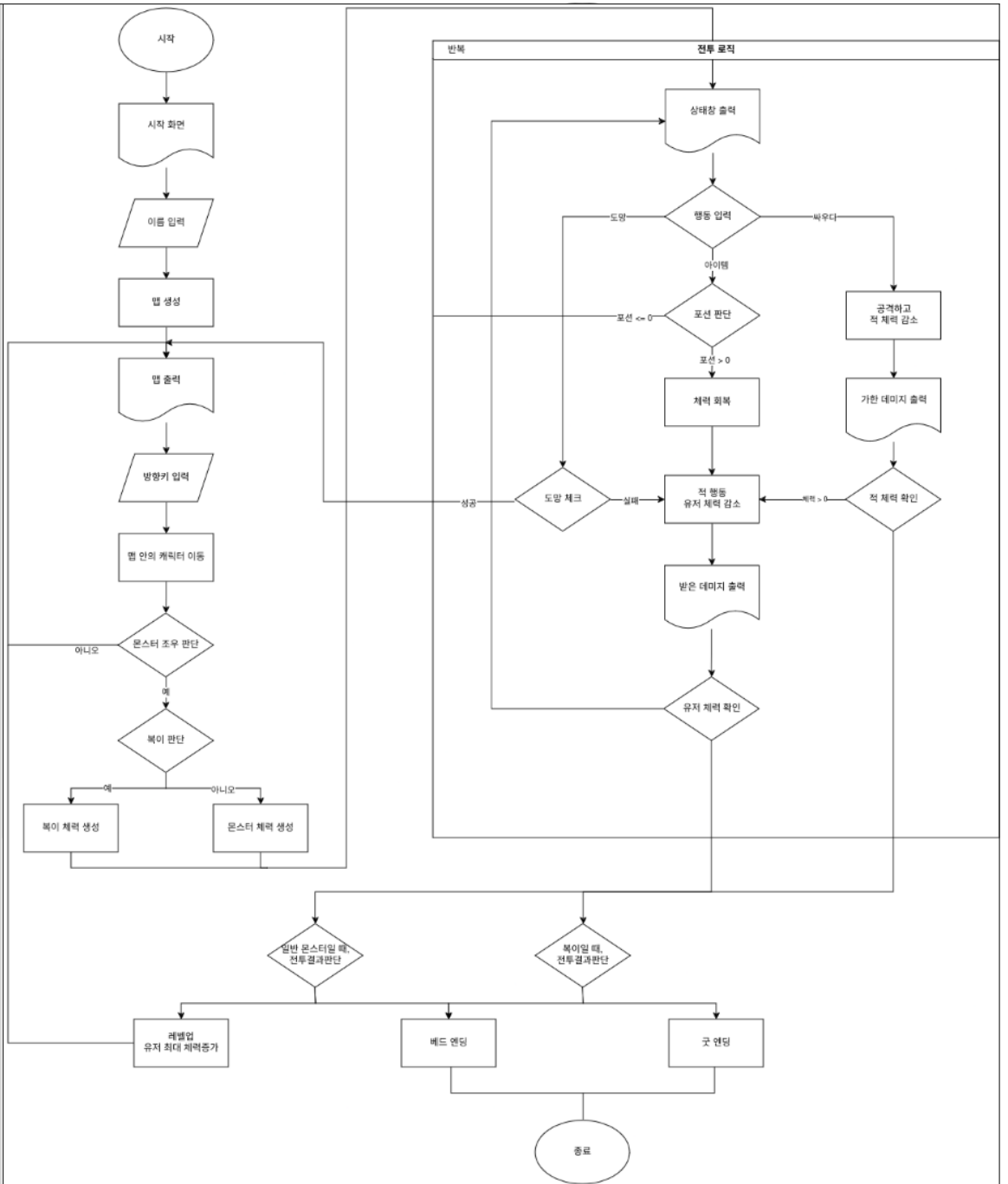
```
[전투 상태창 선택- 2)아이템, 폭탄]
case 3: {
monster_hp -= monster_hp_max/2;
item[2][0]--;
break; }
```

: 몬스터 최대 체력의 50%의
데미지 입힘

코드설명	<pre>[전투 상태창 선택-2)아이템, 도망100%] case 4: { flag = 0; item[3][0]--; break; }</pre>	: flag 0으로 변경해서 도망 체크
	<pre>[전투 상태창 선택- 3)도망] case 3: { randnum = rand() % 100; if(randnum < 30) { flag = 0; }</pre>	: 랜덤숫자 생성 : 30% 확률로 도망 가능 도망 성공시 flag 0으로 변경
	<pre>[몬스터 유저 공격시 - 절대 무적 사용x] if(half_use==0) { randnum = rand() % 100; if(randnum == 0){ randnum_m = (float)(rand() % 11 + 20) / 10; } if(randnum > 0 && randnum < 6) { randnum_m = (float)(rand() % 10 + 10) / 10; } else { randnum_m = (float)(rand() % 5 + 5) / 10; } attack_mon = (int)(hp_max * randnum_m); hp = hp - attack_mon;</pre>	: half_use가 0이면 절대 무적 사용X : 랜덤숫자 생성해서 몬스터 공격력 생성 : 유저 최대체력 * (0.5~3.0배)
	<pre>[몬스터 유저 공격시 - 절대 무적 사용] randnum = rand() % 100; if(randnum == 0) { randnum_m = (float)(rand() % 11 + 20) / 10; } if(randnum > 0 && randnum < 6) { randnum_m = (float)(rand() % 10 + 10) / 10; } else { randnum_m = (float)(rand() % 5 + 5) / 10; } attack_mon = (int)(hp_max * randnum_m); hp = hp - attack_mon/2;</pre>	: half_use가 1이면 절대 무적 사용 : 랜덤숫자 생성해서 몬스터 공격력 생성 : 유저 최대체력 * (0.5~3.0배)

코드설명	<p>[복이 전투 승리]</p> <pre>if(same == 10 && result_battle == 1) { break; }</pre> <p>[몬스터 전투 승리]</p> <pre>if(result_battle == 1){ randnum = rand()% 11; item[0][0] += randnum; randnum = rand()% 100; item[1][0]--; item[3][0]++; randnum = rand()%131 + 120; hp_max *= randnum * 0.01; randnum = rand()%71 + 30; hp += hp_max *(randnum * 0.01); level++; }</pre>	<p>: 몬스터 처치 시 몬스터 이름이 복이면 While문 종료, 게임 승리</p> <p>: 랜덤 확률로 회복포션, 절대무적 포션 획득</p> <p>: 도망 아이템 지급</p> <p>: 랜덤% 수치로 최대체력 증가</p> <p>: 레벨 증가</p>
	<p>[몬스터 전투 패배 시]</p> <pre>else if(result_battle == 2) { break; } printf("패배\n");</pre>	<p>: while문 종료, 게임 종료</p>

순서도



후기

[김 승 수]

이전 프로젝트에서는 분야를 나누기 보다는 모여서 같이 만들었다면, 이번 프로젝트는 담당 파트를 배정받아 만드는 방식으로 진행하였습니다. 일단 담당 파트 배정의 어려움을 알게 되었습니다. 어려울 것이라 예상했던 파트가 쉬울 수도 쉬울 것이라 예상했던 파트가 어려울 수도 있는게 코딩이라는 것을 알게되었습니다. 아니면 제 실력이 부족해서겠지요. 정확한 파트 분배가 가능해 지도록 코드에 대한 이해가 어서 늘어났으면 합니다. 이전 프로젝트에서 디버그를 배워서인지 이번에는 버그가 매우 적게 발생시킬 수 있었지만, 코드를 통합하는 과정의 어려움을 알게 되었습니다. 코드를 이해했더라도 함수화하여 메인 함수 밖으로 빼놓지 않으면 확실히 변경의 과정이 굉장히 어려운 것을 알게 되었습니다. 포인터와 배열을 충분히 이해하여 함수를 자유자제로 사용할 수 있도록 노력해야겠습니다.

[권 철 민]

C언어를 활용한 두 번째 프로젝트를 완료했습니다. c언어가 익숙해졌는지 저번 프로젝트보다 개발 속도가 빨랐고 버그도 적었습니다. 하지만 중간에 추가되는 요구사항을 계속 반영하다 보니 로직이 복잡해지고 코드의 가독성이 떨어졌습니다. 앞으로의 프로젝트는 시작하기 전에 예상되는 문제점을 충분히 고려하고 추가적으로 발생할 수 있는 요구사항도 몇 가지 생각해두고 코드를 짜야겠습니다.

[유 시 온]

이번 프로젝트를 진행하면서 많은 부족함을 느꼈습니다. 로직 부분을 파트로 나눠서 진행하게 되었는데 제가 맡은 부분에 대해 빠른 진행을 하지 못하여 저희 조가 앞 단계로 나아가지 못하였고 로직을 병합을 했을 때 디버깅 과정에서 문제점이 발견되었습니다.

로직을 짜면서 배웠던 부분도 잘 활용하지 못 하였고, 함수를 배웠지만 사용하는 부분에 대해 많이 어려움을 느껴 활용을 못 했습니다.

다음 프로젝트에서는 로직을 읽기 편하고 함수를 사용 할 수 있도록 공부를 해야겠습니다.